STIC-ILL

From: Clow, Lori
Sent: Monday, June 09, 2003 2:35 PM
To: STIC-ILL
Subject: article request

I would like the following articles please:

Journal of Medicinal
Chemistry, American Chemical Society, vol. 38, No. 9, pp. 1431-1436, ·
(Apr. 28, 1995).

Brown, R.D. and Martin, Y.C., "Designing Combinatorial Library Mixtures Using a Genetic Algorithm," Journal of Medicinal Chemistry, vol. 40, No. 15, American Chemical Society, 1997, pp. 2304-2313.

Gillet, V.J. et al., "The Effectiveness of Reactant Pools for Generating Structurally-Diverse Combinatorial Libraries," J. Chem. Inf. Comput.

Kearsley, Simon K. et al., "Chemical Similarity Using Physiochemical Property Descriptors," J. Chem. Inf. Comput. Sci., vol. 36, No. 1, American Chemical Society, 1996, pp. 118-127.

Leland, B.A. et al., "Managing the Combinatorial Explosion," J. Chem. Inf. Comput. Sci., vol. 37, No. 1, American Chemical Society, 1997, pp. 62-70.

Lewis, R.A. et al., "Similarity Measures for Rational Set Selection and Analysis of Combinatorial Libraries: The Diverse Property-Derived (DPD) Approach," J. Chem. Inf. Comput. Sci., vol. 37, No. 3, American Chemical Society, 1997, pp. 599-614.

Martin, E.J. and Critchlow, R.E., "Beyond Mere Diversity: Tailoring Combinatorial Libraries for Drug Discovery," Journal of Combinatorial Chemistry, vol. 1, No. 1, American Chemical Society, Jan. 1999, pp. 32-45.

Agrafiotis, D.K. and Lobanov, V.S., "Ultrafast Algorithm for Designing Focused Combinational Arrays," J. Chem. Inf. Comput. Sci., vol. 40, No. 4, American Chemical Society, Jun. 16, 2000, pp. 1030-1038.

Good, A.C. and Lewis, R.A., "New Methodology for Profiling Combinatorial Libraries and Screening Sets: Cleaning Up the Design Process with HARPick," J. Med. Chem., vol. 40, No. 2, American Chemical Society, 1997, pp. 3926-3936.

Jamois, E.A. et al., "Evaluation of Reagent-Based and Product-Based Strategies in the Design of Combinatorial Library Subsets," J. Chem. Inf. Comput. Sci., vol. 40, No. 1, American Chemical Society, 2000 (pub'd on Web Dec. 9, 1999), pp. 63-70.

Thanks

Lori A. Clow, Ph.D.
Patent Examiner, Art Unit 1631
Crystal Mall 1, Room 12-D-08
Mailbox 12-D-01
(703)-306-5439

Managing the Combinatorial Explosion

Burton A. Leland, Bradley D. Christie, James G. Nourse, David L. Grier, Raymond E. Carhart,
Tim Maffett, Steve M. Welford, and Dennis H. Smith*,†

MDL Information Systems, Inc., 14600 Catalina St., San Leandro, California 94577

Computer software designed to deal with the large amounts of data generated by chemical and biological research programs is described. The software includes methods for representation of the structural components of combinatorial libraries, enumeration of structures and structure-based searches, and comparisons within combinatorial libraries.

INTRODUCTION

A revolution in the rate of generation of chemical and biological information is well under way. This revolution is driven by three factors: (1) the requirement in industry to speed up the process of development of new products which possess specific properties against specific biological targets; (2) the fact that genome projects coupled with high throughput screening (HTS) make it possible to identify targets and assay them quickly; and (3) continuing automation of processes. Robotic methods for HTS and automated methods for chemical synthesis are forcing us to take a new look at how such information is stored and retrieved. Revolutionary approaches to chemical information management are required to keep pace with these developments, especially in the areas of combinatorial chemistry (CC) for construction of large libraries of chemical structures and subsequent association of chemical structural information with chemical inventory, robotics systems, and large databases of test results.

The overall workflow for managing this information includes a cycle beginning with library design and selection and optimization of a chemical synthesis. It continues with steps for structure registration, biological testing, structure determination, and integration of chemical and biological results for SAR studies. The cycle is closed when results are used to define a new experiment involving a new library, for example, in a process for lead optimization.

In this paper we focus on three aspects of the overall workflow which involve management of chemical structures, introducing new methods for (1) representing the structural components of combinatorial libraries and how these representations are linked conceptually to physical samples, reagent inventories, and associated biological test results; (2) enumerating structures if and when required by a specific application; and (3) performing structure-based searches and comparisons over combinatorial libraries.

I. REPRESENTATION—LIBRARIES AND THE "SAMPLE-CENTRIC" VIEW

Chemical vs Combinatorial Libraries. As combinatorial chemistry (CC) and methods for automated synthesis have been adopted by many laboratories, a new lexicon has sprung up to characterize the objects and techniques in this new world. For example, those of us with a chemistry background have often used the term **library** to refer to a set of structures related by a common scaffold and characterized by a Markush representation of the (implied) set of individual structures. The assay laboratories have a different view of "library", however. In this environment, researchers focus on samples, not on chemical structures, and their definition of library is much more general. A library can be any collection of materials from any source. In this sense, a library represents a grouping that has meaning for bioassays and may have no special meaning in terms of chemical structure. For example, a series of extracts from soils queued for testing may be a library. Even when chemical structures can be considered explicitly, a library may still be merely a convenient organizing umbrella beneath which are gathered arbitrary sets of mixtures and specific examples of structures which may have no formal structural relationship. We refer to these libraries as **chemical libraries**. CC represents a special source for samples where there is (generally) an explicit structural relationship among the materials synthesized by CC techniques. For this special case we use the term **combinatorial library**, and Markush structures are one possible, formal representation. Even these terms are arguable. No matter what the jargon, however, this new world is sample-centric, not structure-centric, and keeping this distinction in mind is critical to establishing formal, correct relationships among objects in the world.

In discussing chemical representation we restrict ourselves to combinatorial libraries and their associated structural representations. The techniques required to represent, store, and search the associated Markush structural representations are fundamental components of more general techniques for chemical libraries (which may contain combinatorial components). We recognize but do not address in this paper the fact that any general solution to information management must be able to treat the most general case where arbitrary, unrelated collections of materials can be libraries.

THE SAMPLE-CENTRIC VIEW

Computer-based methods for managing the combinatorial explosion from the perspective of chemical information face an interesting challenge with respect to CC. No longer is there necessarily a 1:1 relationship between a single chemical substance and a test result. HTS process sets of **samples**
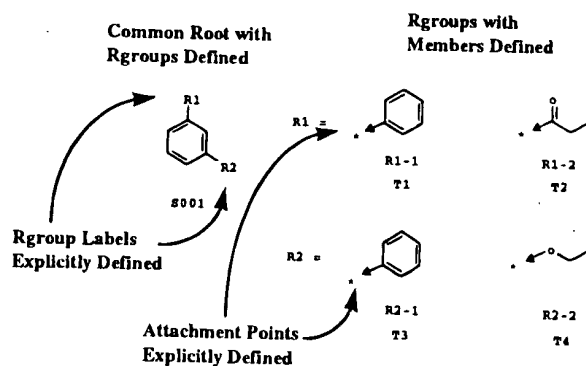

Figure 1. The basic representation of Markush structures. A common scaffold(s) with explicit positions for designated Rgroups is associated with specific sets of defined Rgroup members with explicit attachment points.

which may be grouped into libraries (above). One or more test results are obtained for each sample which, excluding complex extracts of biological materials, may be a component of a chemical or a combinatorial library, the latter as a mixture or as an individual, discrete structure. This **sample-centric** view imposes an additional dimension of complexity in relating samples and test results to objects in computers and database management systems. Here are some of the important distinctions that must be maintained in order to accurately track the workflow of a typical CC application.

Virtual Library vs Samples. The workflow of a CC application often includes two related but distinct chemical groupings, **virtual libraries** and real **samples**. A virtual library is a collection of chemical entities, that is mixtures and/or specific chemical structures, that represent a target library for a proposed synthesis. Such libraries are created *in silico* in order to review and analyze their scope, coverage, diversity, uniqueness, and so forth. A **sample** is an entity that is to be tested, whether prepared by CC or any other technique. The sample is the critical data item in an information management system, and its identifier is what is linked to actual test results. Maintaining this seemingly trivial distinction between "proposed" and actually "tested" is essential in organizing a reliable and accurate workflow.

REPRESENTATION—THE MARKUSH STRUCTURE

Combinatorial libraries are often represented as one or more **Markush** structures. An example is shown in Figure 1. This figure illustrates several of the important requirements for structure representation. There are one or more common "roots", or scaffolds, for a given library; a single scaffold is presented in Figure 1. Each scaffold has explicit points at which Rgroups are to be attached, R1 and R2 in Figure 1. Multiple scaffolds are used to represent variable positions of substitution or to represent libraries resulting from syntheses which produce >1 scaffold. A simple example of the former would be three scaffolds for *ortho, meta, para* disubstituted phenyl rings. The Rgroups each have an arbitrary number of explicitly defined "members", with explicitly defined attachment points, see Figure 1.

Rgroups may be "nested" arbitrarily. In other words, an Rgroup member can itself contain other Rgroups. Nesting can be important for more concise and accurate definition of a library, for example, in "double combinatorial" libraries where a synthetic step which adds an Rgroup member

actually adds a collection of members from another combinatorial library. Finally, the representation accommodates "null" members, in other words, no member at all, among a collection of Rgroup members. Null members represent specific synthetic procedures which purposefully do not add a member for a given Rgroup.

This representation may be contrasted with that developed by the Sheffield group for representing the more general case of generic structures.[1] In their representation, variable positions of attachment are accommodated in a single scaffold, and Rgroup member lists can contain "generic" substituents such as aryl, cycloalkyl, and so forth.

Library and Tag IDs. Extensions to the basic representation include association of specific identifiers, or IDs, with Markush structural components, as illustrated in Figure 2.

The root and each of the Rgroup members are given an ID in the form of text strings, the Root ID and the Rgroup Member IDs, Figure 2. These IDs are used in several of the processes of enumeration, discussed in a subsequent section. A second type of ID associated with Rgroup members is the Rgroup Tag ID, Figure 2. These IDs are used in the process of structure elucidation for the mix-and-split, tagged bead method for library synthesis, see next section.

Representation of "Subgenerics". As described above, it is essential to maintain the correct relationship between a structural object representing a sample and the associated test results obtained for this sample. Often this requires an explicit representation of combinatorial mixtures which are obtained as part of deconvolution experiments designed to identify active components. We term these mixtures "subgenerics". Consider the library **1** in Scheme 1.

One of the possible subgenerics for **1** is library **2**, in Scheme 2. One of the R4 members, O⁻ in **1**, has been selected and connected to the scaffold to yield **2**. Library **2** is itself a representation of a new combinatorial library, but we call it a subgeneric because it is linked internally to its "parent" library **1**. Selecting the methyl group member in R3 in **2** and attaching it to the scaffold yields the new subgeneric **3** in Scheme 3, whose parent is library **2**. This process can continue until one or more specific structures result, for example, **4** and **5**. The process by which subgenerics and specifics are created, in what order, and why it is called **enumeration** and is discussed in detail in a subsequent section.

4   **5**
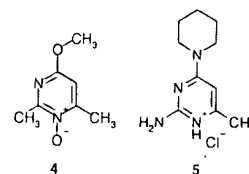
Members vs Reagents. Although Markush structures are generally represented by a scaffold to which sets of Rgroup members are pendant, for example, Figure 1, in the laboratory, such members are appended by chemical reactions which transform reagents into pendant Rgroups as exemplified in Scheme 4. In this restricted sense, members are reagents whose leaving groups have been "clipped" off leading to a computer representation which is amenable to further processing. Again, this is a seemingly trivial distinction. However, it is the reagents which are real world
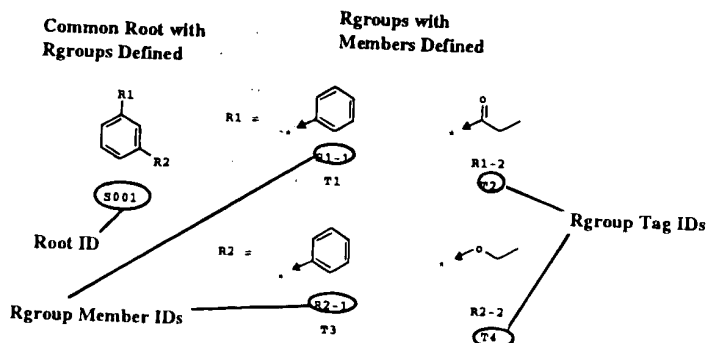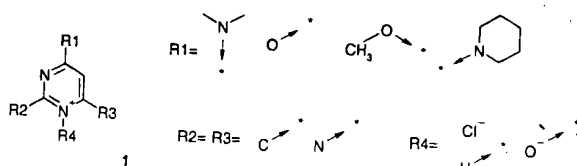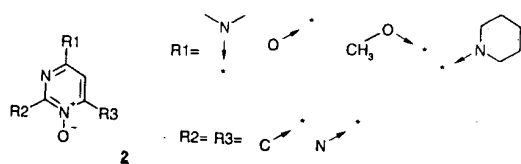
64   *J. Chem. Inf. Comput. Sci., Vol. 37, No. 1, 1997*

LELAND ET AL.

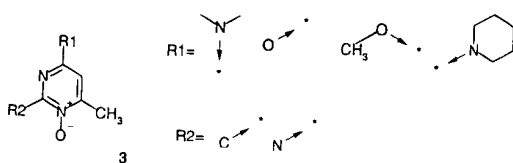**Figure 2.** Extensions to the basic representation to capture Root and Rgroup member IDs and Rgroup tag IDs.

Scheme 1

Scheme 2

Scheme 3

Scheme 4

A Benzodiazapine Library

90 Specific Structures

MW = 236.2757 - 452.4737

Figure 3. A benzodiazepine library representing 90 discrete structures.
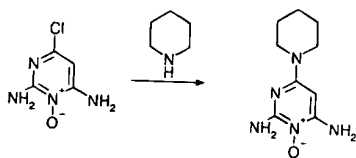
objects, no matter how many transformations into members have been carried out in the computer or laboratory. As real chemicals, reagents must be selected, ordered, inventoried, associated with safety information, and often tracked cradle to grave within an organization. It is essential in an information management system to retain explicit linkages from a member to its source reagent and from that reagent to whatever additional tracking systems are in place for managing chemical inventories.

Combinatorial vs Selected Products. Libraries based on combinatorial synthesis, where the *possible* products can be represented as a Markush structure with defined Rgroups, *e.g.*, R1, R2, R3, ..., Scheme 1, are experimentally of two types. The first we refer to as strictly **combinatorial** where all possible products are targets for synthesis. For example, the simple benzodiazepine library[2] shown in Figure 3 represents 90 specific structures targeted for synthesis. In this case, the number of products, as specific structures, is a

strict mathematical product $n \times m \times o \times ...$ where $n,m,o,...$ are the numbers of members of each of the respective Rgroups; $5 \times 3 \times 3 \times 2 = 90$.

Often, however, not all possibilities are actually synthetic targets. A preprocessing step may consider the diversity of the reagents or the complete structures, as in a virtual library. Based on this analysis a **selected** subset of possible products may be chosen. This subset may be a strict product as above, but one which uses only selected members from each Rgroup. Alternatively, the subset may not be a mathematical product at all and may represent just a selection of possible products. These distinctions are critical in associations among libraries, structures, samples, and data. If some form of enumeration is used to construct computer representations of products for subsequent database storage, search, and retrieval, fine control over the enumeration is essential when the goal is to produce structures which represent samples for assay.

II. SUM VS PRODUCT ALGORITHMS

Before describing enumeration and structure searching in more detail, we discuss the importance of sum vs product algorithms. In CC applications, several forms of constrained enumeration and several forms of structure searching for identity, substructure, stereochemistry, overlap, similarity, and so forth must consider all chemical objects in a database. This includes all libraries each of which may contain both

MANAGING THE COMBINATORIAL EXPLOSION

J. Chem. Inf. Comput. Sci., Vol. 37, No. 1, 1997 **65**

Table 1. Scope of Combinatorial Libraries[a]

library	no. of structures represented
(structure)	3.2×10^6
(structure)	64×10^6
(structure)	10^3
(structure)	10^6
(structure)	10^{12}

[a] From 20 amino acids or 1000 reagents for each Rn.

Table 2. Overall Efficiency—Sum vs Product

pentapeptide
$\sum n_i = 10^2$
$\prod n_i = 3.2 \times 10^6$

discrete and Markush structures. Libraries containing Markush structures, whether they represent mixtures or are a convenient shorthand for a set of subgenerics or specifics, pose several problems related to computational complexity; these can be *large* search problems. Some simple examples are presented in Table 1. As this table indicates, relatively simple combinations of scaffolds and Rgroup member lists lead quickly to large numbers of structures. If one considers double combinatorial libraries, where for example one or more of the Rn members is itself a combinatorial library, then extraordinary numbers of structures are possible. This is a classic product vs sum problem. Consider the pentapeptide of Table 1. Table 2 illustrates the potential advantages of algorithms treating the sum of Rgroup members, 100 in this case, vs those which treat the product.

Systems for enumeration and searching must be capable of applying all standard methods directly to Markush structures to avoid combinatorial explosions. Either the Markush structures represent actual samples, so are the appropriate chemical objects to be searched, or they represent such a vast number of specific structures that it is impractical to build all the specifics and then search them. But dealing directly with Markush representations is only a part of the solution. Equally important are algorithms which have sufficiently high performance to be useful in real applications.

We have attacked several enumeration and searching problems for CC using the strategy of identifying algorithms that scale as the **sum** of the number of structural components of a Markush structure (for example, as measured by the total number of atoms in the scaffold plus all members of all Rgroups), rather than as the **product** (for example, as measured by the total number of atoms in the entire set of (potentially) enumerated specific structures). Simple examples of this strategy are implicit in the methods used for enumeration based on data-driven constraints (below), where, for example, molecular weight and formula constraints are implemented based on calculations of maximum and mini-

Bulk - a construction problem

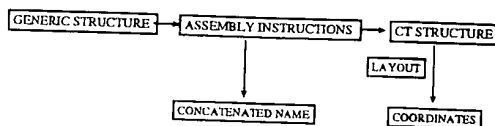Constrained - a combined search and construction problem

Figure 4. A schematic of the basic enumeration strategy. Enumeration can proceed to assembly instructions or to full connection tables, the latter followed by a layout step to obtain useful coordinates for 2D or 3D processing. Enumeration may be "bulk" or constrained, as described in the text.

mum numbers of component properties using the components themselves rather than first assembling them into complete structures and then doing the calculations. Additional examples are given in the sections below.

III. ENUMERATION

Enumeration is the process of constructing computer representations of subgeneric or specific structures from a given Markush structure, as illustrated schematically in Figure 4.

"Bulk" enumeration is the brute-force process of constructing representations of all possible subgenerics or specifics from a given Markush structure. Constrained enumeration provides alternative mechanisms for using structural properties or Rgroup constraints to focus on a subset of possible results.

The representations of structures may be either assembly instructions or computable structural representations as connection tables ("CT" in Figure 4), where the former can be used at a later time to derive the latter. Assembly instructions are simply, for each subgeneric or specific structure, a concatenated string of Root and Rgroup member IDs (Figure 2) from which a connection table can be constructed.

The basic enumeration process itself is trivial and would remain uninteresting were it not for the special requirements of (A) constraints on the enumeration driven by the CC application itself; (B) constraints on an enumeration used to aid in selecting subsets or in identifying structures; and (C) computable 2D and 3D structural representations, "LAYOUT" in Figure 4, for registration, searching, reporting, and consumption by a variety of additional methods for chemical information and computational chemistry.

Turning first to requirements (A) and (B), we identify two aspects of the combinatorial chemistry application and workflow that provide constraints on enumeration, **process-driven** (A) and **data-driven** (B) constraints. Both can provide assembly instructions or computable structural representations, the latter including derived coordinates, for example, using a **layout** procedure (requirement C).

A. Process-Driven Constraints. Process-driven constraints result from the nature of the specific CC experiment carried out in the laboratory. Mirroring that experiment *in silico* may require special constraints on the enumeration so that the objects created in the computer track the experimental results in the laboratory.

1. Assembly Instructions. In order to track structural objects including roots, or scaffolds, and their associated

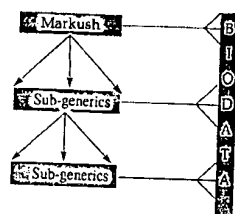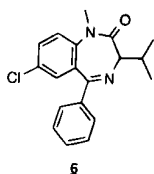I. Iterative deconvolution of mixtures by resynthesis

Figure 5. Constrained enumeration applied to CC experiments where structure determination is based on iterative deconvolution of mixtures. Selected Rgroups are enumerated in succession to reflect the actual experiment, yielding subgenerics which represent actual mixtures to be tested. These mixtures may be tested in several bioassays, so there is a one-to-many relationship between each subgeneric and its associated biodata.

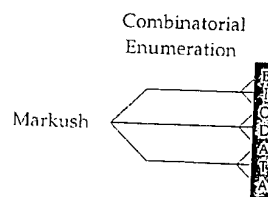II. Synthesis of spatially-arrayed discretes

Figure 6. Constrained enumeration applied to CC experiments where structure determination is based on synthesis of specific, or discrete, structures, located by plate/well IDs. Combinatorial or selective enumeration may be used to build the set of structures corresponding to the synthesis. Each structure may be tested in several bioassays, so there is a one-to-many relationship between each discrete structure and its associated biodata.

Rgroup members and to track in turn the relationship of the members to reagents and inventory, identifiers are associated with each structural object. A simple approach to enumeration is, for each enumerated structure, to construct a unique name that is the concatenation of the root, or scaffold, and appropriate member names to yield a string of characters. For example, for the benzodiazepine library of Figure 3, the character string **BZA-iPr-Ph-Cl-C** is the assembly instruction for **6**. We refer to these strings as assembly instructions

6

because within the scope of a given Markush structure which details the scaffold(s), members, and points of attachment, the strings are sufficient to drive later construction of an actual structure on demand. In addition, the assembly instructions are in a form compatible with robotic synthesizers and can be used to direct subsequent automated synthesis.

2. Mixtures. Markush representations of combinatorial libraries, their association with subgenerics and specific structures, and the relationship to samples and test results were introduced previously. If a laboratory experiment involves **mixtures** with subsequent structure identification based on deconvolution,[3] the various subgeneric structures in a sequence such as Figure 5 will represent explicit samples and will have associated assay data (Figure 5). The enumerator must be constrained to construct the subgenerics in a sequence mirroring that in the laboratory. This means that specific Rgroups must be selectable for enumeration at each step in a well-specified series of enumerations that tracks the identification of active samples through the deconvolution process. For each selected Rgroup, generally one at each step of deconvolution, its Rgroup members are substituted in turn during enumeration, leaving the remaining Rgroups on the scaffold.

3. Discretes by Combinatorial Enumerations. A second approach to CC involves experiments using specific, or discrete, structures, with identification of structures in active samples based on the synthesis process, plate/well identifiers, or tagging strategies.[3] In this application, an enumerator must be instructed to build the appropriate specific structures using combinatorial or selective enumera-

tion. In this section we discuss use of combinatorial enumeration to produce a set of discrete structures; selective enumerations are discussed in the next section. In either approach the Markush representation is a useful shorthand used in the enumeration and subsequent searching but is not an actual sample in the laboratory.

As discussed above, a **combinatorial enumeration** results in discrete structures whose number is a multiplicative product of the Rgroups and members selected for enumeration. Two methods must be allowed to mirror an actual experiment. The first method uses all members of each selected Rgroup; such "bulk" enumerations are often used, for example, to construct a complete set of discrete structures for a virtual library (see above) or to represent the actual targets for synthesis, Figure 3. The second method allows selection among individual Rgroup members, for each Rgroup, which are to participate in the enumeration. The result is still a multiplicative product of the selected Rgroups/members but is often a significantly reduced set. This approach is used, for example, when a scientist reviewing members for selected Rgroups finds one or more members (or their associated reagents) which are incompatible with the experimental conditions may not be available in inventory, may have chemical properties incompatible with an assay, and so forth. The relationship between the structures and their associated assay data is summarized in Figure 6.

4. Discretes by Selective Enumeration. An experimenter may choose a subset of structures for enumeration, for example, based on manual analysis, random selection, or selection based on diversity analysis of a virtual library (see above). Such subsets are generally not a strict combinatorial enumeration. If the results of the selection are in the form of assembly instructions, then these instructions are used to drive the enumerator to produce actual structures which are to be synthesized and associated with assay data. This is implemented as a data-driven constraint as described in the next section.

B. Data-Driven Constraints. Several methods for structure elucidation are used to identify discrete, active compounds or active compounds in mixtures including, for example, mass spectrometric analysis of materials, electron capture GC of tags, or substructural information from other experimental data. Each piece of data is a constraint on the identity of a specific structure(s) in a sample. Each such **data-driven** constraint can be phrased as a constraint on an enumeration. Thus, each of the constraints listed below is applied during a constrained enumeration process which

MANAGING THE COMBINATORIAL EXPLOSION

J. Chem. Inf. Comput. Sci., Vol. 37, No. 1, 1997 **67**

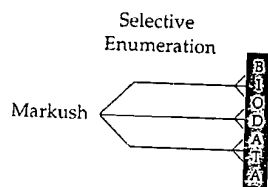III. Encoded, mix-and-split synthesis

Figure 7. Constrained enumeration applied to CC experiments involving (1) encoded, mix-and-split synthesis, in which case tag IDs are used to drive the enumeration to produce the selected structure or (2) selective enumeration where assembly instructions based on Root and Rgroup member IDs are used to construct only the specified set of structures.

constructs one or more single structures which obey the constraint(s). In each case the structures which do not satisfy the constraints are never constructed.

1. Molecular Weight or Formula. Mass spectrometric analysis of a sample may yield information on molecular weight or molecular formula or ranges of the same. The enumerator is programmed to make efficient use of the constraint to build *only* the structures which meet the constraint based on the Markush representation itself. In other words, the program does *not* need to build each structure, then test to see if it meets the constraint. The program performs a constrained tree search computing the sum of amu values for the scaffold and individual Rgroup members.

Another very useful application of a molecular weight constraint is to constrain a library so that all specific structures are below or within a given molecular weight or range. Because solubility, partition, and transport are all related indirectly to molecular weight, often an upper limit, for example 500 or 600 amu, is set for a library. As a data-driven constraint, setting molecular weight limits is a very efficient way to construct a desirable set of discrete structures given a Markush structure and represents another alternative to constructing structures that will mirror the actual experiment.

2. Tag Ids or Assembly Instructions. Tags are unique molecular entities synthesized on beads together with the synthesis of a unique chemical structures in mix-and-split approaches to synthesis of combinatorial libraries.[3] Sensitive techniques such as ECGC are used to identify a code representing the tags. In turn, the codes can be used to drive the enumeration process which produces the single structure corresponding to the code. A similar approach can be followed using assembly instructions. Input of an assembly instruction—the string concatenating the scaffold and member names—results in construction of the single structure corresponding to that name. Input of a table of assembly instructions results in construction of the set of structures noted in the section **Discretes by Selective Enumeration**, above. The relationship of structures and biodata for this type of experiment is summarized in Figure 7. Structure construction by IDs is performed using the Markush representation itself and is another implementation of a sum as opposed to a product algorithm.

3. Substructure Constraint. Given substructural information on an unknown sample, for example from NMR, it is possible to use the enumerator to construct the specific structures which contain the observed substructure. This data-driven constraint is more problematic than those de-scribed above. Often the scientist is looking for verification (does this Markush structure contain the substructure within any of its implied structures, yes or no) rather than actual structure construction. If verification is desired, then substructure search, described below, is a better way to answer this question. Even if construction is desired in principle, it may not be desirable in practice. Often the number of results is surprisingly large, and we are still searching for the right implementation that addresses the real problem at hand. For example, our current approach is to verify and construct some representative examples.

C. Layout. The final phase of an enumeration process is construction of a graphical representation of the subgeneric or specific structures specified by the constraints. User requirements are clear. Users desire aesthetically pleasing 2D or 3D structure diagrams, where the former must maintain the orientation of the scaffold of the Markush from which they came to aid visual review, construct useful SAR tables, and so forth. The ultimate destinations for such structures are chemical databases, electronic/printed reports which maintain chemical intelligence, and other computational techniques, for example, SAR analysis or modeling.

We use two methods for laying out structures. The first recomputes the 2D coordinates from a connection table. The second assembles the structure using the coordinates of the scaffold and the members. In both cases the stereochemistry of the structure is carefully noted and after layout is corrected if necessary. In both cases the final orientation of the structure is determined by using the original orientation of the scaffold.

IV. SEARCHING

In this section we focus on some of the problems and solutions for various types of structure searching among libraries. There are many other important issues with respect to differentiation of virtual libraries vs real samples and integrated searching among structures and associated assay data which must be kept in mind if the goal is to have a truly integrated system, but these are beyond the scope of this text.

Solutions to representation and searching for generic chemical structures in patents have been presented in a series of papers from Professor Lynch and his co-workers.[1] Other groups have also made important contributions to this problem.[4] However, as with enumeration (above), there are several search-related problems in CC that are special to the nature and requirements of the application itself. For example, there are many precise properties which can be derived from the precise Markush structural representations. These properties can be used to solve several problems in enumeration (for example, data-driven molecular weight and formula constraints, above) and searching (for example, see **Full Structure** search, below). Further, the application requires several interesting extensions to basic search technologies. These extensions are potentially applicable to the patent problem as well but have never been addressed to our knowledge: (1) stereochemical searching, critical to establishing reliable structure/function relationships in biological systems; (2) overlap of; and (3) similarity of combinatorial libraries, both important in addressing the how two or more libraries are related. These extensions are discussed in more detail below.
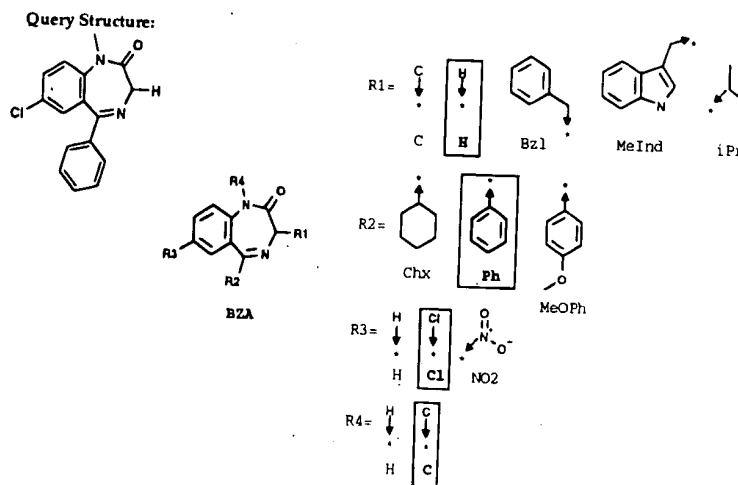
68   J. Chem. Inf. Comput. Sci., Vol. 37, No. 1, 1997

LELAND ET AL.

Figure 8. Results of a substructure search using the indicated query posed against the single Markush structure representing the benzodiazepine library of Figure 3. The hit includes the root **BZA** and one member, boxed, from each of the Rgroups.

We distinguish two fundamental approaches to structure searching, path tracing algorithms, and set reduction algorithms. Path tracing algorithms generally map a query into a structure atom by atom. The mapping may be repeated partially or entirely for different Rgroup members and will often backtrack. All combinations of Rgroup members may need to be considered. Set reduction algorithms consider all possible query atom to structure atom pairs and iteratively eliminate those that cannot be mapped.[1,5]

A. Basic Requirements. The following is a list of capabilities for structure-related searches which are important for the CC application. All such searches must be capable of using a Markush or specific representation for a query which is posed over a target set of Markush or specific structures, representing one or more combinatorial libraries, and restricted to a specified library if required. It is convenient to consider four separate cases depending on whether the query or the structure is Markush or discrete. The discussion that follows is applicable to both full structure and substructure search, including stereochemical searches.

1. Discrete Query vs Discrete Structure Target. This is conventional structure search where the query may possess many diverse atom and bond query properties. This case has been well studied and implemented many times.[6]

2. Discrete Query vs Markush Structure Target. The question posed here is whether the query substructure can be found in *any* of the structures implied by the Markush structure target. This search is done on the Markush structure itself; there is no requirement for enumeration. The query structure must be able to map into any part of the scaffold or Rgroups. Once a member of an Rgroup is mapped, all other members of that Rgroup are disallowed for further mapping. The mapping will imply one or more discrete structures implied by the Markush structure, and these may need to be constructed. This method is an another example of use of a sum as opposed to a product algorithm. A simple, illustrative example is shown in Figure 8. Although the query was a substructure search, for this particular query against this library, Figure 8, the results of an exact match search would be the same.

3. Markush Query vs Discrete Structure Target. The question posed here is whether *any* of the set of structures summarized by the Markush query matches a given discrete

structure. Again, this is conventional structure search with Markush queries. In general the scaffold is mapped first, and then the Rgroups are considered individually. This means that the atoms of a second Rgroup in the query can map to the same atoms as were mapped by the first Rgroup. For example, in structure **8** the divalent oxygen of the anhydride is mapped by oxygen atoms in both Rgroups in the query **7** with this stipulation. This is another example of a sum vs product efficiency since the mapping does not need to backtrack through all the possible combinations of Rgroup members.[7]
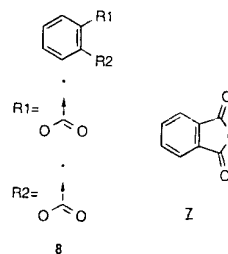
4. Markush Structure Query vs Markush Structure Target. The question posed here is whether *any* of the set of structures summarized by the Markush query matches *any* of the structures implied by the Markush structure target. This search is done on the Markush structure itself using a set reduction approach; there is no requirement for enumeration. All individual processing steps are done by effectively going through the full set of scaffold and Rgroup atoms so the process is sum-based. If there is more than one mapping possible, the final result will be a set of query atom-structure atom pairs that only implies the actually mappings. Actual construction of all the possible mappings is product-based.[8] An application of this algorithm is illustrated below in the section **Library Comparison**.

Another important application of this technique is full structure (identity, or exact match) search. A Markush query can match only a Markush target structure, and a match occurs only if all structures implied by the query match all structures implied by the target. Our method prescreens the search using several properties derived from the Markush structures themselves, for example, number of specific structures, minimum, maximum, average molecular weight,

MANAGING THE COMBINATORIAL EXPLOSION

J. Chem. Inf. Comput. Sci., Vol. 37, No. 1, 1997 **69**

and so forth. The screening methods only depend on the sum of the total number of atoms. Full structure search establishes identity of Markush structures working solely with the Markush representations, *whether or not the scaffolds and lists of Rgroup members are the same.*[8] We use the Markush structure query vs Markush structure target method described above to verify the identity.

B. Extended Requirements. The CC application requires that it be possible to answer additional questions relating to scaffolds or specific members. For example, it is often important to know whether a proposed scaffold has been used for syntheses of combinatorial libraries in the past, and what were the results of analysis for structures built on that scaffold. It is also important to know if given members (as members or reagents) have been used before, what were the results, and is the reagent still in inventory. Therefore, it is important to ask questions restricted to scaffolds or members in Markush structural targets.

1. Target Markush Restricted to Scaffold. Queries match only if the query, or one of its implied structures in the case of a Markush query, is entirely contained within a scaffold of a Markush target. This comparison is performed on the Markush target itself. The result is a list of Markush structures in the database that contain the query as an explicit scaffold.

2. Target Markush Restricted to Rgroup Members. Queries match only if the query or one of its implied structures in the case of a Markush query is entirely contained within one member of a Markush target. This comparison is performed on the Markush target itself. The result is a list of Markush structures in the database that possess a member that matches the query.

C. Library Comparison. Another special characteristic of the CC application is the necessity for comparing libraries. As more libraries are acquired from outside sources, are assembled from existing compound databases, and are synthesized by combinatorial methods, avoiding costly rework becomes an important objective. The rapid growth of numbers of libraries has created a situation which is very analogous to that of managing proprietary corporate databases, where analysis of this historical record is essential to ensure proposed work is indeed novel.

Establishing novelty in the context of historical databases of individual substances, including discrete structures, polymers, mixtures, and so forth, has been accomplished with identity (full structure) searches and similarity searches. The former of course establishes whether a substance has been recorded (registered) previously. The latter explores whether similar substances have been recorded. Which method is used depends on the application. Similarly for libraries, there are two relevant methods for comparison whose use again depends on the application, **overlap** and **similarity**.

1. Overlap. Overlap is a measure of how many structures two libraries have in common, with structural identity used as the method of comparison. This method is used when decisions must be made on exactly which materials to purchase or to synthesize and is clearly analogous to checking a proposed new compound against a proprietary (or public) database in conventional chemical information systems. Overlap is determined explicitly based on comparison of two libraries (more generally and most often, comparison of a proposed library with a database containing many libraries). We have developed algorithms that perform this comparison

Scheme 5

L1

Scheme 6

L2
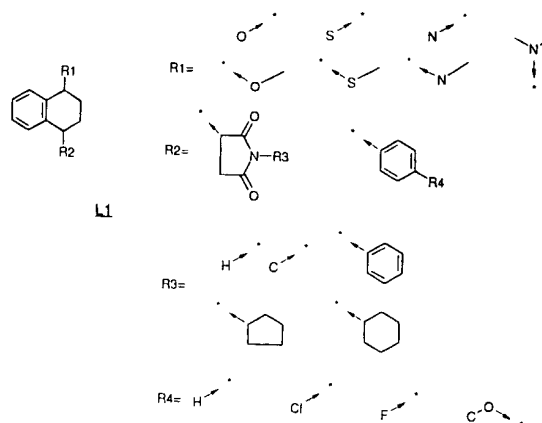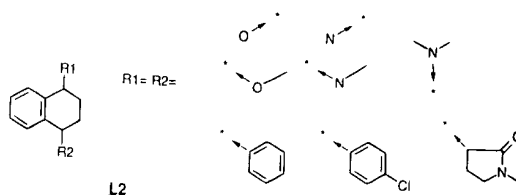
with *no* requirement to enumerate any Markush structures in the libraries. A key algorithm is one which compares two Markush structures, detects that there are common structures, computes a metric on degree of overlap, and enables direct construction of common (or different) structures.

Consider the following example which has been simplified by making the scaffolds equivalent; the method outlined does *not* require equivalence of scaffolds. Assume that the combinatorial library **L1** shown in Scheme 5 has been proposed for purchase or synthesis.

Assume that an existing database contains a combinatorial library **L2** represented by Scheme 6.

The library **L1** represents 63 structures **L2** represents 64 structures. There are 10 structures in common between the two libraries, which is 16% of both **L1** and **L2**. Two of the ten structures are **9** and **10**. These metrics can be calculated from the search results without enumerating the libraries.

9

10
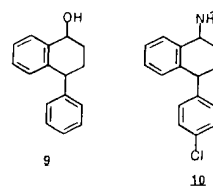
2. Similarity. Similarity is a measure of how alike are two libraries, with some measure of structural similarity among the two sets of structures used as the method of comparison. This approach is most often used during analysis of virtual libraries, where measures of similarity and diversity are important in establishing the value of such libraries. A paper presented at this conference has outlined a method for obtaining substructure keys, which can be used to compute similarities or to perform clustering, directly from a Markush representation.[9]

V. CONCLUSIONS

We have discussed several important problems and illustrated solutions, in representation, enumeration, and searching of combinatorial libraries. We have shown how these solutions are linked to experimental procedures including relationships to bioassay data. We have been successful in implementation of several algorithms which scale as the sum as opposed to product of the number of atoms in a representation. Although more work remains to be done, practical problems can now be managed easily in integrated chemical and biological information systems.

REFERENCES AND NOTES

(1) Holliday, J. D.; Lynch, M. F. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 17. Evaluation of the Refined Search. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 659−662.

(2) Dewitt, S. H.; Kiely, J. S.; Stankovic, C. J.; Schroeder, M. C.; Reynolds Cody, D. M.; Pavia, M. R. "Diversomers": An Approach to Non-peptide, Nonoligomeric Chemical Diversity. *Proc. Natl. Acad. Sci. U.S.A.* **1993**, *90*, 6909−6913.

(3) Gallop, M. A.; Barrett, R. W.; Dower, W. J.; Fodor, S. P. A.; Gordon, E. M. Applications of Combinatorial Technologies to Drug Discover. 1. Background and Peptide Combinatorial Libraries. *J. Med. Chem.* **1994**, *37*, No. 9, April 29.

(4) *Computer Handling of Generic Chemical Structures*; Barnard, J. M., Ed.; Gower: Hampshire, 1984.

(5) Barnard, J. M. Substructure Searching Methods: Old and New. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 532−538.

(6) Christie, B. D.; Leland, B. A.; Nourse, J. G. Structure Searching in Chemical Databases by Direct Lookup Methods. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 545−547.

(7) Wipke, W. T.; Nourse, J. G.; Moock, T. Generic Queries in the MACCS System. In *Computer Handling of Generic Chemical Structures*; Barnard, J. M., Ed.; Gower: Hampshire, 1984; pp 167−178.

(8) Nourse, J. G., et al. Manuscript in preparation.

(9) Downs, G. M.; Barnard, J. M. *J. Chem. Inf. Comput. Sci.* In press.

CI960088T